

JANUARY 1986

LIDS-P-1528

DELAYS IN ACYCLICAL DISTRIBUTED DECISIONMAKING ORGANIZATIONS\*

by

Victoria Y-Y Jin\*\*

Alexander H. Levis\*\*

Pascal A. Remy\*\*

ABSTRACT

Distributed decisionmaking organizations with synchronous protocols are represented using Petri nets. An algorithm for computing time delays is developed. Starting with a matrix representation of the organizational structure, all possible information processing paths are scanned and the time delay associated with each one is computed. When the decision strategies are known, the expected delay of the overall system can be obtained. An alternative formulation based on the flow matrix description of the Petri net is also presented.

---

\*This work was carried out at the MIT Laboratory for Information and Decision Systems with support provided by the Office of Naval Research under Contracts N00014-84-K-0519 (NR 649-003) and N00014-85-K-0782 (NR 964-001).

\*\*Laboratory for Information and Decision Systems, MIT, Cambridge, MA 02139.

Report Documentation Page			Form Approved OMB No. 0704-0188		
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE <b>JAN 1986</b>		2. REPORT TYPE		3. DATES COVERED <b>00-01-1986 to 00-01-1986</b>	
4. TITLE AND SUBTITLE <b>Delays in Acyclical Distributed Decisionmaking Organizations</b>			5a. CONTRACT NUMBER		
			5b. GRANT NUMBER		
			5c. PROGRAM ELEMENT NUMBER		
6. AUTHOR(S)			5d. PROJECT NUMBER		
			5e. TASK NUMBER		
			5f. WORK UNIT NUMBER		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) <b>Massachusetts Institute of Technology, Laboratory for Information and Decision Systems, 77 Massachusetts Avenue, Cambridge, MA, 02139-4307</b>			8. PERFORMING ORGANIZATION REPORT NUMBER		
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)			10. SPONSOR/MONITOR'S ACRONYM(S)		
			11. SPONSOR/MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION/AVAILABILITY STATEMENT <b>Approved for public release; distribution unlimited</b>					
13. SUPPLEMENTARY NOTES					
14. ABSTRACT					
15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES <b>6</b>	19a. NAME OF RESPONSIBLE PERSON
a. REPORT <b>unclassified</b>	b. ABSTRACT <b>unclassified</b>	c. THIS PAGE <b>unclassified</b>			

## DELAYS IN ACYCLICAL DISTRIBUTED DECISIONMAKING ORGANIZATIONS\*

Victoria Y-Y Jin  
Alexander H. Levis  
Pascal A. Remy

Laboratory for Information and Decision Systems  
Massachusetts Institute of Technology  
Cambridge, Massachusetts 02139, USA

### ABSTRACT

Distributed decisionmaking organizations with synchronous protocols are represented using Petri nets. An algorithm for computing time delays is developed. Starting with a matrix representation of the organizational structure, all possible information processing paths are scanned and the time delay associated with each one is computed. When the decision strategies are known, the expected delay of the overall system can be obtained. An alternative formulation based on the flow matrix description of the Petri net is also presented.

### KEYWORDS

Organization theory, Petri nets, Distributed decisionmaking.

### 1. INTRODUCTION

Decisionmakers in a distributed organization have access to specified information sources and control some specified resources. Usually, even in simple organizations, there are many alternative paths through which information can be processed. Internal decisions, which affect the sharing of information and the issuance of directives or commands, determine the paths information will take. In a distributed decisionmaking organization, each member makes his own internal decisions on the basis of decision rules which depend on the task, his role in the organization, his style, and his workload.

A useful measurement of organizational performance is the time interval between the moment data about an event are received by the organization and the moment a response can be made. This time delay reflects the organization's ability to respond to external events in a timely manner. In earlier work (Boettcher and Levis, 1983) another performance measure, the accuracy of the response, was used to evaluate and compare organizations. Together, accuracy and timeliness of response provide a better basis for evaluating organizational structures.

The focus of this paper is on the computation of the time delay that characterizes a given organizational form. A distributed decisionmaking organization (DDMO) is often a large scale system which contains many decisionmakers (DMs) and decision support systems (DSSs) that interconnect them. To evaluate the organization's response

time, all possible information processing paths must be identified and then the delay associated with each path computed. An algorithm for (a) scanning all paths and identifying concurrent paths in acyclical organizations, (b) computing the conditional probability of each path given the decision strategies of the organization members, and (c) computing path delays and the expected delay has been developed and implemented. This algorithm is applicable when the protocols that specify the sequence of events within the organization and the admissible interactions between members are synchronous.

In the next section, Petri nets are introduced to represent organizations forms. In the third section, the algorithm for scanning all paths is described and the analytical formulation of the problem is given. Once the paths have been identified, the computation of the path delays is direct (section 4). The analytical and computational approaches described in this paper are illustrated with a two-person organization; the results have also been applied to the comparison of a two three-person organization -- a hierarchical and a parallel one.

### 2. PETRI NET REPRESENTATION OF ORGANIZATIONAL FORMS

Petri nets were used by Tabak and Levis (1985) to represent organizational forms because they show explicitly the interactions between decisionmakers and the sequence of operations within an organization. The basic elements of a Petri net, a directed bipartite graph (Peterson, 1981) are the circle node or place that represents a signal, and the bar node or transition that represents a process or function. The actual presence of information in a place is denoted by one or more tokens in that place. A transition is enabled, if there are tokens in all its input places. It was convenient to extend the basic Petri net formalism to include a special type of transition to represent the internal decisionmaking by an organization member. In the model of the decision-maker (Levis, 1984) the internal decisionmaking is represented by a switch and a decision rule that specifies the position of the switch. In the case of a DM, this means which algorithm (from a finite set) is to be used to process the data at some stage.

The concept of transitions being enabled allows the explicit representation of organizational protocols that dictate under what conditions decisions may be made or information processed. The decision switch with its associated rules allows the introduction of decision rules that determine which one of several enabled transitions (or processes) is to be activated or fired.

The Petri net representing a simple two-person DDMO (Boettcher and Levis, 1983) is shown in Figure 1.

\*This work was carried out at the MIT Laboratory for Information and Decision Systems with support provided by the Office of Naval Research under Contracts N000014-84-K-0519 (NR 649-003) and N00014-85-K-0782 (NR 964-001).

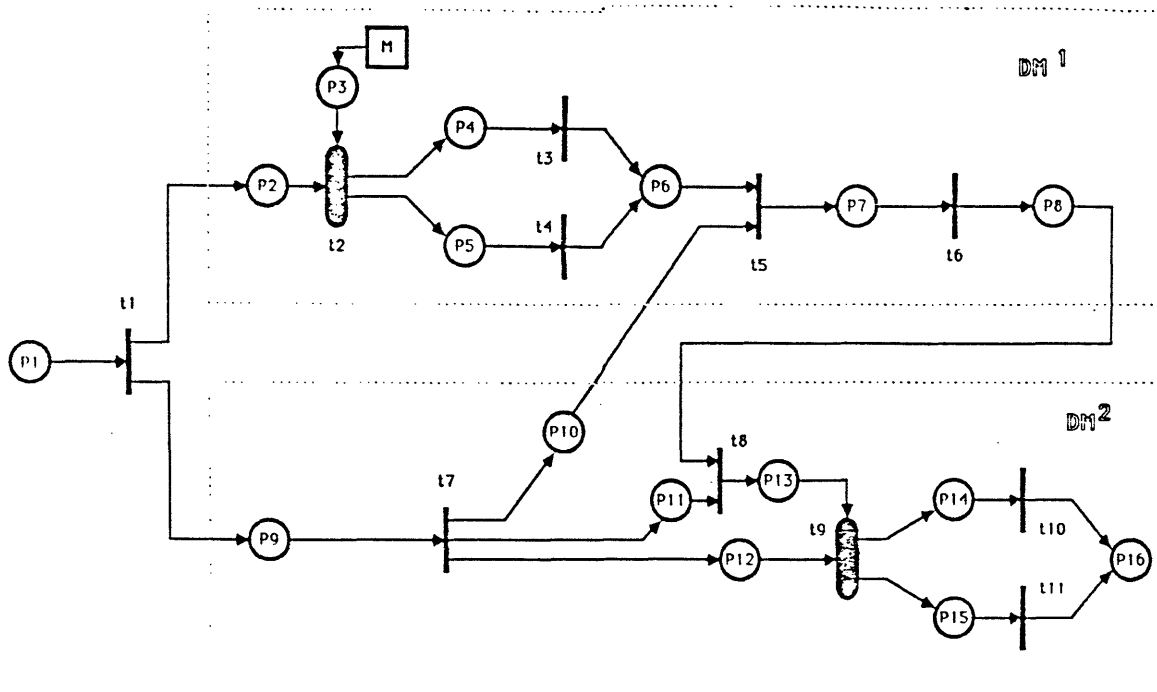


Figure 1. Petri net representation of a two-person organization

The data about an event, denoted by place  $p_1$ , are partitioned by transition  $t_1$ :  $DM^1$  receives data  $p_2$  about the event while  $DM^2$  receives  $p_9$ . A decision switch,  $t_2$ , controlled by decision  $p_3$  with its rule stored in memory  $M$ , determines which algorithm,  $t_3$  or  $t_4$ , will be used to process the data and produce the situation assessment  $p_6$ . However, processing cannot continue, according to the protocol of this organization, until  $DM^2$  receives the assessment  $p_{10}$  from  $DM^1$  which is produced by a single situation assessment algorithm represented by transition  $t_7$ . The two situation assessments,  $p_6$  and  $p_{10}$ , are fused together in the information fusion (IF) transition  $t_5$ ; the output is the revised situation assessment  $p_7$ , which is processed then by the single response selection algorithm  $t_6$  to produce  $DM^1$ 's response  $p_8$ . In this case, the output does not act on the environment, but is a command to the second decisionmaker. The situation assessment of  $DM^2$ ,  $p_{11}$ , combines with the command  $p_8$  to produce, through the command interpretation (CI) transition  $t_8$ , the decision  $p_{13}$ , that controls the decision switch  $t_9$ , which, in turn, chooses the response selection algorithm that is to process the situation assessment,  $p_{12}$ . (Note that  $p_{10}$ ,  $p_{11}$ , and  $p_{12}$  represent the same signal). The signal  $p_{12}$  is processed either by  $t_{10}$  or  $t_{11}$  to produce the organization's output,  $p_{16}$ .

Synchronization occurs at three points in this organization. The input data,  $p_1$ , arrive at the same time to both DMs; the processing is synchronized at the information fusion transition  $t_5$ , and, once again, at the command interpretation transition  $t_8$ . This feature of Petri nets, namely, the explicit depiction of synchronization points, is one reason they are very useful in representing organizational protocols.

This very simple organizational form could be used to describe the interrelationship between a central and a regional office, or a corporate headquarters and one of its many franchises. Market data, national and local, are received by headquarters ( $DM^1$ ) and the local operation ( $DM^2$ ), respectively. The local operation provides processed information about local conditions to headquarters; the latter, on the basis of both national and local

information, issues guidelines (or formulates corporate policy) that are then communicated to the local operations. The latter, in turn, selects its own response to local market conditions, but within the guidelines established by headquarters.

If there were no decision switches, then the Petri net representing an organization could be represented in terms of a flow matrix  $\Delta$  (Peterson, 1981). Let  $i$  denote a place and  $j$  denote a transition. Then, the element  $\Delta_{ij}$  of the flow matrix is defined as follows

$$\Delta_{ij} = \begin{cases} -1, & \text{if } p_i \text{ is an input place to transition } t_j \\ 0, & \text{if } p_i \text{ and } t_j \text{ are not connected directly} \\ 1, & \text{if } p_i \text{ is an output place of } t_j \end{cases}$$

The dimension of such a matrix is  $n \times m$  where  $n$  is the number of places and  $m$  is the number of transitions.

A Petri net with decision switches can be represented by a set of Petri nets without decision switches and by the set of decision rules that control the switches. Each one of these reduced nets is obtained by setting each one of the switches at a specific setting - thus eliminating the switching action. If there are  $R$  switches, each with  $s_r$  positions, then there are

$$\rho = \prod_{r=1}^R s_r$$

reduced Petri nets in the set. For example, the set for the organization of Fig. 1 contains four reduced nets. One is obtained by setting switch  $t_2$  to select transition  $t_3$ , and switch  $t_9$  to  $t_{11}$ ; the reduced net is shown in Figure 2 and its corresponding flow matrix,  $\Delta^1$ , in Table 1.

If the switch  $t_2$  is set to select transition  $t_4$ , then the resulting matrix  $\Delta^4$  has the same numerical values as  $\Delta^1$ , but place 5 replaces place 4 in the

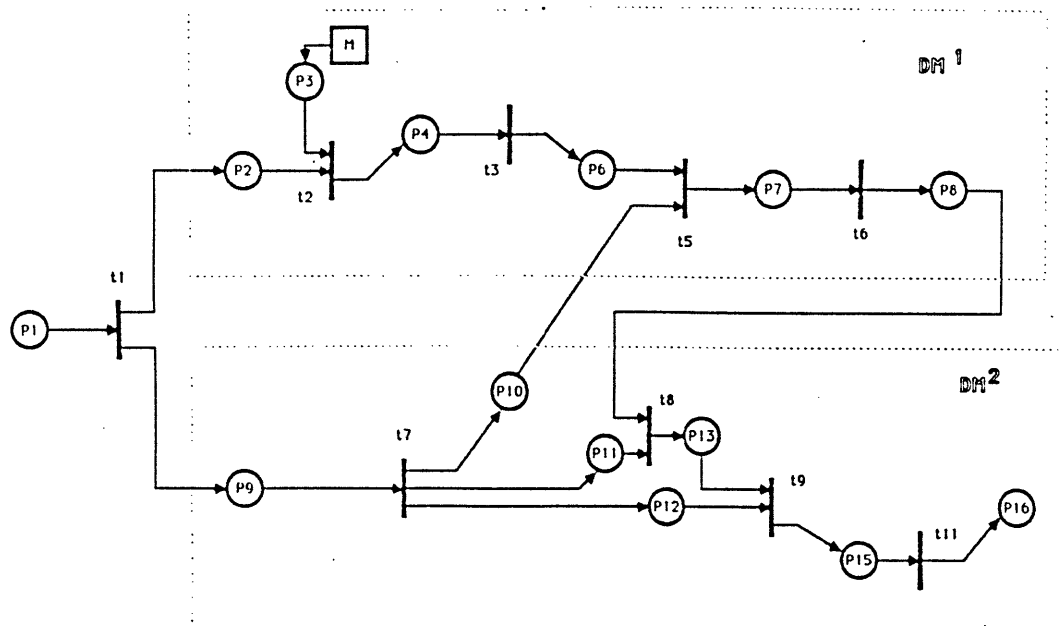


Figure 2. Reduced net #2

TABLE 1 Flow Matrix  $\Delta^2$

p \ t	1	2	3	5	6	7	8	9	11
1	-1	0	0	0	0	0	0	0	0
2	1	-1	0	0	0	0	0	0	0
3	0	-1	0	0	0	0	0	0	0
4	0	1	-1	0	0	0	0	0	0
6	0	0	1	-1	0	0	0	0	0
7	0	0	0	1	-1	0	0	0	0
8	0	0	0	0	1	0	-1	0	0
9	1	0	0	0	0	-1	0	0	0
10	0	0	0	-1	0	1	0	0	0
11	0	0	0	0	0	1	-1	0	0
12	0	0	0	0	0	1	0	-1	0
13	0	0	0	0	0	0	1	-1	0
15	0	0	0	0	0	0	0	1	-1
16	0	0	0	0	0	0	0	0	1

fourth row and transition  $t_4$  replaces transition  $t_1$  in the third column.

It is possible to construct an alternative representation of the Petri net in terms of an interconnection matrix,  $C$ , that is based on transitions (columns) and on links (rows) rather than places. The elements  $C_{ij}$  of this interconnection matrix are defined as follows:

$$C_{ij} = \begin{cases} -1, & \text{if link } i \text{ leaves transition } j \\ 0, & \text{if link } i \text{ is not connected to transition } j \\ +1, & \text{if link } i \text{ arrives at transition } j \end{cases}$$

The matrix is constructed row by row by considering each link that leaves a transition (-1) and indicating the transition to which it is directed (+1). Since each row represents a single link, it must have exactly one +1 and one -1 and, consequently, the sum of its elements must be zero. This is not the case, of course, for the links that connect input places (sources) to transitions and the links that connect transitions to output places, or sinks. The former are recognized by a row with a single +1, while the latter by a row with a single -1. These properties of the inter-

connection matrix are very useful in constructing tests for capturing data errors. The interconnection matrix for the two-person organization of Fig. 1 is shown in Table 2.

### 3. PATH IDENTIFICATION

The algorithm that is based on the interconnection matrix is presented first (Jin, 1985). The problem is to find all possible paths from an input place (source) to an output place (sink). The transition that follows immediately the source place becomes the main root of the tree that represents the paths; every path forms a branch of this tree.

TABLE 2 The Interconnection Matrix  $C$

$\lambda \backslash t$	1	2	3	4	5	6	7	8	9	10	11
1	1	0	0	0	0	0	0	0	0	0	0
2	-1	1	0	0	0	0	0	0	0	0	0
3	0	1	0	0	0	0	0	0	0	0	0
4	0	-1	1	0	0	0	0	0	0	0	0
5	0	-1	0	1	0	0	0	0	0	0	0
6	0	0	-1	0	1	0	0	0	0	0	0
7	0	0	0	-1	1	0	0	0	0	0	0
8	0	0	0	0	-1	1	0	0	0	0	0
9	0	0	0	0	0	-1	0	1	0	0	0
10	-1	0	0	0	0	0	1	0	0	0	0
11	0	0	0	0	1	0	-1	0	0	0	0
12	0	0	0	0	0	0	-1	1	0	0	0
13	0	0	0	0	0	0	-1	0	1	0	0
14	0	0	0	0	0	0	0	-1	1	0	0
15	0	0	0	0	0	0	0	0	-1	1	0
16	0	0	0	0	0	0	0	0	-1	0	1
17	0	0	0	0	0	0	0	0	0	-1	0
18	0	0	0	0	0	0	0	0	0	0	-1

#### The Algorithm

The elements of  $C$  can be interpreted as follows:

- If  $C_{ij} = -1$  and  $C_{ik} = 1$ , then transition  $t_j$  precedes and is connected to transition  $t_k$ .
- If there are more than one (-1) in a column  $j$ , then transition  $t_j$  is a root or a subroot of the tree.
- If there are  $q$  (+1) in a column  $j$ , then  $q$  paths coalesce after they reach transition  $t_j$ .

flow matrices  $\Delta^i$ , as described in Section 2. The above procedure, the determination of the minimal support T-invariants, must be applied to each one of the matrices  $\Delta^i$ . In this simple case, the null space of each  $\Delta^i$  is one-dimensional and, consequently, there is only one firing sequence that transfers a token from  $p_1$  to  $p_{16}$  in each reduced net. For example, for the reduced net of Fig. 2 and the flow matrix of Table 1, the only minimal support T-invariant is

$$j = 1 \ 2 \ 3 \ 5 \ 6 \ 7 \ 8 \ 9 \ 11 \\ \xi^2 = [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]$$

Each minimal support T-invariant of  $\tilde{T}$  represents a complete path. In this example, each  $\Delta^i$  yields one complete path for a total of four for  $\tilde{T}$ .

This procedure is very efficient for determining complete paths (a set of concurrent simple paths). If it is desired to determine the individual simple paths, from a given source to a given sink then the S-invariants of each  $\tilde{T}$  must be determined. Again (Memmi and Roucairol, 1979) the minimal support invariants of  $\Delta^i$  are sufficient to determine the simple paths.

For the example of Fig. 2, there are four minimal support S-invariants with 1 at the first place and the sixteenth place ( $p_1$  and  $p_{16}$ ). They are,

$$i = 1 \ 2 \ 3 \ 4 \ 6 \ 7 \ 8 \ 9 \ 10 \ 11 \ 12 \ 13 \ 15 \ 16 \\ \mu_1^1 = [1 \ 1 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1] \\ \mu_2^1 = [1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1 \ 1 \ 0 \ 0 \ 0 \ 1 \ 1 \ 1] \\ \mu_3^1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1] \\ \mu_4^1 = [1 \ 0 \ 0 \ 0 \ 0 \ 0 \ 0 \ 1 \ 0 \ 0 \ 1 \ 0 \ 1 \ 1]$$

If each marking representing a simple path is expressed as a set of places, then the complete path is the union of the four sets that correspond to  $\mu_i^1$ ,  $i = 1, 2, 3, 4$ . There are four simple paths that correspond to each  $\Delta^i$ , therefore there are 16 simple paths and four complete paths. However, only 10 of these simple paths are distinct. This is the same result obtained using the algorithm based on the interconnection matrix C.

#### 4. CALCULATION OF TIME DELAYS

The algorithm for the computation of delays is based on the following two propositions (Jin, 1985). Consider the  $m$ -th subpath ending in transition  $t_i$  and let the delay associated with a transition  $t_k$  be denoted  $d_k$ . Then the delay of the subpath  $\{(i, m)\}$  is

$$D_{im} = \sum_k d_k \quad (3)$$

where  $k$  ranges over the indices of the transitions in the subpath. Let  $D_{i1}$ ,  $D_{i2}$ , and  $D_{i3}$  be the delays of three subpaths,  $\{(i, 1)\}$ ,  $\{(i, 2)\}$ , and  $\{(i, 3)\}$  ending at transition  $t_i$ . Then the delay up to that transition is the maximum of the three delays, i.e.,

$$D_i = \max (D_{i1}, D_{i2}, D_{i3}) \quad (4)$$

Consider again the two-decisionmaker organization. There are three subpaths that end at transition  $t_i$ ,  $\{(5, 1)\}$ ,  $\{(5, 2)\}$ , and  $\{(5, 3)\}$  as shown in Table 3. If the delay at each transition is equal to a unit of time,  $\tau$ , and if the decision switch is assumed to have negligible delay, then

$$D_{51} = D_{52} = D_{53} = 3\tau$$

and the delay from  $t_1$  to  $t_i$  is also equal to  $3\tau$ .

The delay of a simple path consisting of  $n$  subpaths is the sum of the delays of the subpaths. For example, from Table 3, simple path  $Z_2$  is composed of subpaths  $\{(9, 2)\}$  and  $\{(11, 1)\}$ . Then the delay of  $Z_2$  is given by

$$D_2 = D_{9,2} + D_{11,1} = [(1+1+0) + (1+0) - 0]\tau = 3\tau$$

where 0 is subtracted, formally, because the decision switch is counted twice. The delay of a complete path that consists of concurrent simple paths, is the delay of the simple path with the maximal delay. For example, the complete path of Fig. 2 which consists of simple paths  $Z_1$ ,  $Z_2$ ,  $Z_3$ , and  $Z_4$ , has delay equal to  $6\tau$ , the delay of path  $Z_1$  or  $Z_4$ .

After all delays are computed, the paths with minimal time delay and those with maximal delay are easily identified. Often, however, the expected delay is used to compare alternative organizational forms.

To calculate expected delay, probabilities associated with each path need to be calculated first. Probabilities are usually given as conditional probabilities associated with each transition. If transition  $t_i$  has only one input from the previous transition,  $t_j$ , then the conditional probability  $p(t_i/t_j)$  is 1. If  $t_i$  is a transition of a  $n$ -way decision switch  $t_j$ , a set of conditional probabilities  $(t_i/t_j) \leq 1$  will be assigned such that

$$\sum_{i=1}^n p(t_i/t_j) = 1 \quad (5)$$

The probability that information processing will follow a certain simple path  $X$  with  $n$  transitions is given by

$$p(X) = p(t_1) \prod_{i=2}^{n-1} P(t_i|t_{i-1}) \quad (6)$$

where  $X$  is the path number  $j$ ,  $t_i$  is some transition on the path, and  $t_{i-1}$  is the transition preceding  $t_i$ . Table 4 shows the conditional probability matrix  $P_j$  for the organization of Fig. 1. Table 5 shows the probabilities associated with the simple paths and the complete paths for this organization. The expected delay can be calculated by the following equation:

$$E_r = \sum_{i=1}^r p_i D(i) \quad (7)$$

where  $p_i$  and  $D(i)$  are the probability and time delay associated to the  $i$ -th complete path and  $r$  is the total number of complete paths in a system.

There are 4 complete paths  $j$ ; path  $\xi_1$  has delay of  $6\tau$  and probability of 0.372. In this particular example, because the delay of all complete paths is  $6\tau$ , the expected delay is  $6\tau$ .

TABLE 4 Probability Matrix  $P_3$  of Example

$P_3 =$	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.6	0.4	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	1.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.3	0.7
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.7
	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0

TABLE 5 Probabilities Associated With Simple Paths and Complete Paths

Simple Path	Conditional Probability
$Z_1$	$1 \times 1 \times 1 \times 0.7 = 0.7$
$Z_2$	$1 \times 1 \times 0.7 = 0.7$
$Z_3$	$1 \times 1 \times 1 \times 0.3 = 0.3$
$Z_4$	$1 \times 1 \times 0.3 = 0.3$
$Z_5$	$1 \times 1 \times 1 \times 1 \times 1 \times 0.3 = 0.3$
$Z_6$	$1 \times 0.4 \times 1 \times 1 \times 1 \times 1 \times 0.3 = 0.12$
$Z_7$	$1 \times 0.6 \times 1 \times 1 \times 1 \times 1 \times 0.3 = 0.18$
$Z_8$	$1 \times 1 \times 1 \times 1 \times 1 \times 0.7 = 0.7$
$Z_9$	$1 \times 0.4 \times 1 \times 1 \times 1 \times 1 \times 0.7 = 0.28$
$Z_{10}$	$1 \times 0.6 \times 1 \times 1 \times 1 \times 1 \times 0.7 = 0.42$
Complete Path	Probability
$\xi_1$	0.372
$\xi_2$	0.558
$\xi_3$	0.028
$\xi_4$	0.042

## 5. APPLICATIONS

Two three-person organizational forms have been used to test the approach and the algorithm for computing delays. Both organizations have to perform the same set of tasks: monitor an airspace and handle objects that are within it. The first organization, the parallel one, illustrates distributed decisionmaking. The airspace is divided into three sectors with each decisionmaker assigned to one sector. Each DM monitors his sectors and makes decisions about the objects in it. Since the objects can cross from one sector to another (e.g., in air traffic control, aircraft move from one sector to another) decisionmakers #1 and #3 who monitor the outside sectors must communicate (share information) with the middle DM, #2. No DM issues commands or restricts the options of the others.

In the hierarchical organization, the airspace is divided into two sectors, with each one assigned to a single DM. Since the resulting workload will be high for each DM, a central region is defined that straddles the two sectors. A supervisor is introduced who does not observe the airspace directly, but receives information about objects in the central region from both DMs. He processes the received information (not raw data) and allocates these objects to either one of the two DMs (command inputs) depending on the trajectory of the object. This would represent the action of a supervisor who resolves conflicts and allocates some of the load so that neither of the subordinates is overloaded.

The interconnection matrix for each organizational form was constructed. The parallel organization had sixteen simple paths while the hierarchical had twenty. Because of the interconnections between organization members and the existence of decision

switches, the parallel organization has 128 distinct complete paths. The corresponding number of complete paths for the hierarchical organization is sixty four.

As a result of choosing a constant time delay of one unit for each process or transition and zero delay for the decision switches, the delay of each complete path of the parallel organization is six units, while the delay of the paths of the hierarchical organization is eight units. Consequently, the expected delays are six and eight units, respectively.

The detailed analysis of this application can be found in Jin (1985).

## 6. CONCLUSION

In this paper, Petri nets have been used for modeling distributed decisionmaking organizations with synchronous protocols. The problem of computing organizational delays was formulated and two approaches were described for its solution. One is based on an algorithm and an associated interconnection matrix, while the other is based on the flow matrix and the associated S and T-invariants. Which approach is preferable depends on the number of decisions switches and the number of positions of each switch. For a small number of simple switches, the analytical method is effective and yields additional insight. For more complex organizations, the algorithmic approach is preferable.

Current research is addressing (a) the problem of computing delays when asynchronous protocols are used and (b) the design of asynchronous protocols to meet delay specifications.

## 7. REFERENCES

- Boettcher, K. L., A. H. Levis (1983). Modeling and analysis of teams of interacting decisionmakers with bounded rationality. *Automatica*, 6, 703-709.
- Jin, V. Y-Y (1985). Delays for Distributed Decisionmaking Organizations. Report LIDS-TH-1459, Laboratory for Information and Decision Systems, MIT, Cambridge, MA.
- Levis, A. H. (1984). Information processing and decisionmaking organizations: a mathematical description. *Large Scale Systems*, 7, 151-163.
- Memmi, G. G. Roucairol (1979). Linear algebra in net theory. In W. Brauer (Ed.), *Lecture Notes in Computer Science* (No. 84) - Net Theory and Applications. Springer-Verlag, Hamburg, 213-223.
- Peterson, J.L. (1981). *Petri net theory and the modeling of systems*. Prentice-Hall, Englewood, Cliffs, NJ.
- Tabak D., A. H. Levis (1985). Petri net representation of decision models. *IEEE Trans. on Systems, Man, and Cybernetics*, Vol. SMC-15, No. 6, 812-818.